**Operating System**

**Common Interview Questions**

**1. What is an operating system, and what are its primary functions?**

Operating System is a Cumulative program which connects computer user to computer Hardware providing Resource management, process Management, I/O management, Memory management.

**2. Explain the difference between process and thread.**

Process is an instance of Running Program whereas thread is individual part of a process running separately.

|  |  |  |
| --- | --- | --- |
|  | Process | Thread |
| 1 | Process is a program in Execution | Thread is small executing segment in process |
| 2 | Process can be with long time duration | Threads are quick |
| 3 | Process has its own memory | Threads share memory |
| 4 | Process are independent of each other | Threads can be dependent on each other causing errors |
| 5 |  |  |
|  |  |  |
|  |  |  |

**3. What is virtual memory, and how does it work?**

**Virtual memory is a logical memory separate from the physical memory created in in storage when physical memory is full or requirement for large memory is there.**

**I such scenario only small part of program is needed to be in physical memory for execution so the other major part is stored in a logical memory which is created sharing address space.**

**4. Describe the difference between multiprogramming, multitasking, and multiprocessing.**

**5. What is a file system, and what are its components?**

**File System is a Program which manages files. In Linux everything is treated as file.**

**And everything is arranged in heiraichal order starting from (/).**

**Components of File System are diffent types of file**

* **Normal file: these files can be text, Binary(Image, video) or it can be executable.**
* **Directory file: these files reperent directories(folders). Which contain normal files or other directories called sub directories.**
* **Device File: every device connected is treated as file from mouse to printer,etc.**

**6. What is a deadlock, and how can it be prevented?**

**7. Explain the difference between a kernel and a shell.**

**Kernel is a core program of operating System where as Shell is lexternal layer of operating system exposed to the user.**

**Shell is used to write and send commands to the OS. Whereas Kernel executes thes instrucions by activating related programs.**

**8. What is CPU scheduling, and why is it important?**

**9. How does a system call work?**

**10. What is the purpose of device drivers in an operating system?**

**11. Explain the role of the page table in virtual memory management.**

**12. What is thrashing, and how can it be avoided?**

**13. Describe the concept of a semaphore and its use in synchronization.**

**14. How does an operating system handle process synchronization?**

**15. What is the purpose of an interrupt in operating systems?**

**16. Explain the concept of a file descriptor.**

**17. How does a system recover from a system crash?**

**18. Describe the difference between a monolithic kernel and a microkernel.**

**19. What is the difference between internal and external fragmentation?**

**20. How does an operating system manage I/O operations?**

**21. Explain the difference between preemptive and non-preemptive scheduling.**

**22. What is round-robin scheduling, and how does it work?**

**23. Describe the priority scheduling algorithm. How is priority assigned to processes?**

**24. What is the shortest job next (SJN) scheduling algorithm, and when is it used?**

**25. Explain the concept of multilevel queue scheduling.**

**26. What is a process control block (PCB), and what information does it contain?**

**27. Describe the process state diagram and the transitions between different process states.**

**28. How does a process communicate with another process in an operating system?**

**29. What is process synchronization, and why is it important?**

**30. Explain the concept of a zombie process and how it is created.**

**31. Describe the difference between internal fragmentation and external fragmentation.**

**32. What is demand paging, and how does it improve memory management efficiency?**

**33. Explain the role of the page table in virtual memory management.**

**34. How does a memory management unit (MMU) work?**

**35. What is thrashing, and how can it be avoided in virtual memory systems?**

**36. What is a system call, and how does it facilitate communication between user programs and the**

**operating system?**

**37. Describe the difference between a monolithic kernel and a microkernel.**

**38. How does an operating system handle I/O operations?**

**39. Explain the concept of a race condition and how it can be prevented.**

**40. Describe the role of device drivers in an operating system.**

**41. What is a zombie process, and how does it occur? How can a zombie process be prevented?**

**42. Explain the concept of an orphan process. How does an operating system handle orphan**

**processes?**

**43. What is the relationship between a parent process and a child process in the context of process**

**management?**

**44. How does the fork() system call work in creating a new process in Unix-like operating systems?**

**45. Describe how a parent process can wait for a child process to finish execution.**

**46. What is the significance of the exit status of a child process in the wait() system call?**

**47. How can a parent process terminate a child process in Unix-like operating systems?**

**48. Explain the difference between a process group and a session in Unix-like operating systems.**

**49. Describe how the exec() family of functions is used to replace the current process image with a**

**new one.**

**50. What is the purpose of the waitpid() system call in process management? How does it differ from**

**wait()?**

**51. How does process termination occur in Unix-like operating systems?**

**52. What is the role of the long-term scheduler in the process scheduling hierarchy? How does it**

**influence the degree of multiprogramming in an operating system?**

**53. How does the short-term scheduler differ from the long-term and medium-term schedulers in**

**terms of frequency of execution and the scope of its decisions?**

**54. Describe a scenario where the medium-term scheduler would be invoked and explain how it helps manage system resources more efficiently.**